



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

Realización de los servicios web para una aplicación móvil de apuestas deportivas

Autor:

David SERRANO HERNÁNDEZ

Supervisor:

Sergio AGUADO GONZÁLEZ

Tutor académico:

Dolores María LLIDÓ ESCRIVÁ

Fecha de lectura: 18 de Julio de 2016

Curso académico 2015/2016

Resumen

Este documento describe el proceso de desarrollo del backend de la aplicación móvil MyBets para Android e iOS. MyBets es una aplicación que permitirá a los usuarios realizar apuestas sin ánimo de lucro. Utilizando una metodología iterativa e incremental se ha ido refinando la aplicación e incrementando su funcionalidad hasta obtener unos servicios web REST y un portal web completamente funcionales.

El desarrollo de la aplicación se ha realizado durante la estancia en prácticas en la empresa Soluciones CuatroOchenta, empresa especializada en el desarrollo de aplicaciones.

La aplicación conjuntamente con este documento conforman el trabajo de final de grado en el grado de ingeniería informática de la Universitat Jaume I.

Palabras clave

Web, ágil, REST, Symfony

Keywords

Web, agile, REST, Symfony

Índice general

1. Introducción.....	5
1.1. Contexto y motivación del proyecto	5
1.2. Objetivos del proyecto.....	6
2. Descripción.....	7
2.1. Descripción general	7
2.2. Descripción funcional	7
2.3. Alcance	8
3. Planificación	11
3.1. Metodología	11
3.2. Planificación	11
3.3. Estimación de recursos y costes del proyecto	15
3.4. Planificación real del proyecto.....	16
4. Análisis y diseño del sistema.....	19
4.1. Análisis de datos	19
4.2. Diseño de la arquitectura	23
4.3. Diseño de la interfaz	28
5. Implementación	33
5.1. Tecnologías y herramientas utilizadas	33
6. Desarrollo proyecto	35
6.1 Sprint 1.....	35
6.2. Sprint 2.....	36
6.3. Sprint 3.....	38
6.4. Sprint 4.....	40
6.5. Sprint 5.....	41
6.6. Sprint 6.....	43
6.7. Sprint 7.....	45
6.8. Sprint 8.....	48
7. Conclusiones.....	51
7.1. Resumen.....	51
7.2. Objetivos logrados	51

7.3. Valoración personal	51
A. Estudio detallado de servicios web desarrollados y JSON de envío y respuesta.....	55
B. Glosario.....	83

Capítulo 1

Introducción

El proyecto MyBets es desarrollado por el alumno durante su estancia en la empresa *CuatroOchenta*. En este capítulo se realiza una descripción de la empresa, del producto que se desea construir y sus objetivos.

1.1. Contexto y motivación del proyecto

1.1.1. Contexto actual

La empresa Soluciones CuatroOchenta, de aproximadamente 50 trabajadores, se dedica a la realización de aplicaciones a medida a la demanda de sus clientes. La empresa cuenta con más de 50 aplicaciones desarrolladas durante mas de sus 4 años de experiencia.

Esta empresa destaca por su filosofía de centrar el desarrollo de sus proyectos en la integra colaboración real con el cliente y usuarios. Tanto es así que, actualmente, la empresa cuenta con una aplicación llamada CuatroOchenta interactive, que permite que el cliente cree las pantallas para que la empresa realice el proyecto con esas características.

Para ello la empresa cuenta con profesionales de diversos campos como el de diseño gráfico, marketing, comunicación, programación web, programación en iOS, programación en Android e incluso un departamento de videojuegos.

1.1.2. Motivación

Actualmente vivimos en una sociedad muy globalizada donde las personas intercambian información en todo momento. La cantidad de usuarios de Internet se ha disparado en los últimos años. Según datos recientes actualmente 3419 millones de personas tienen acceso a Internet. En enero del 2015, el 42% de las personas tenía acceso a Internet [1].

Este fenómeno ha incrementado el uso de aplicaciones móviles, como la del proyecto a tratar en esta ocasión, MyBets, y es por eso que actualmente conseguir que la comunicación fluya de forma rápida y segura es un requisito indispensable para tener éxito. Para conseguir que la información de todos estos usuarios sea debidamente gestionada se utilizaran servicios web.

Un servicio web es una tecnología cuya finalidad consiste en permitir la comunicación entre 2 o más dispositivos a través de la World Wide Web. En los servicios web se utiliza el protocolo HTTP y permite intercambiar datos en formato XML y JSON [2].

Por todo esto, desarrollar un proyecto con este tipo de tecnología resulta de gran interés debido a su posible uso en aplicaciones de distintos ámbitos.

1.2. Objetivos del proyecto

La principal meta de este proyecto es implementar unos servicios web y un portal web completo para la administración del sistema. Los servicios web se encargaran de dar servicio a las aplicaciones en Android e IOS que otros estudiantes están desarrollando en paralelo durante la realización de este proyecto.

Por otra parte, también se tiene que poblar la base de datos mediante una interfaz cómoda adaptada al sistema (usuarios, retos, categorías, competiciones, apuestas, resultados,...) cuya función cumple el portal web de la administración. El objetivo principal de esto es permitir que los diferentes equipos que participan en el proyecto puedan acceder a la base de datos sin hacerlo manualmente.

Por otro lado, el desarrollo de este proyecto tiene como objetivos formativos que el alumno encargado de desarrollar la aplicación adquiera las siguientes competencias:

- Aprender acerca de la programación web y su combinación con otras tecnologías
- Trabajar cooperativamente en el desarrollo de un proyecto de software en un ámbito real.
- Aprender a utilizar y dominar tecnologías y herramientas que se utilizan en el mundo profesional.

Capítulo 2

Descripción

2.1. Descripción general

El proyecto MyBets es una aplicación de apuestas deportivas sin dinero en juego donde puedes retar a tus amigos a jugar contra ti y apostar sobre los resultados de todo tipo de eventos deportivos.

El proyecto es desarrollado para la empresa CuatroOchenta, situada en las instalaciones Espaitec 2 dentro del campus de la universidad. Las apuestas se realizan sobre modalidades de competiciones de distintos deportes.

El proyecto se ha dividido en 3 módulos: el desarrollo de una aplicación para el sistema Android, el desarrollo de una aplicación para iOS y el desarrollo de los servicios web del servidor que se encargaran de gestionar las peticiones de las 2 aplicaciones anteriores, junto a un portal web para el administrador para facilitar la gestión de la información del sistema. Mi labor será la de desarrollar este último módulo.

2.2. Descripción funcional

En la aplicación MyBets los jugadores podrán invitar a sus amigos a la aplicación y retarles a jugar partidas en grupo para apostar sobre el resultado de alguna competición registrada en el sistema. Si el jugador acierta, recibirá puntos que pueden ser consultados por sus amigos.

Una competición es un conjunto de jornadas que están formadas por diversos partidos. En estos partidos, los equipos se enfrentan y el jugador debe tratar de acertar el resultado del encuentro.

Los juegos son creados por el administrador a partir de una competición y en él se define la modalidad de apuesta que se utilizara de las disponibles para esa competición.

Las partidas de los juegos pueden estar formadas por varias personas a los que solo se pueden acceder si eres invitado y estas son creadas por el jugador. Para ello en el momento de creación de la partida el jugador podrá escoger a que amigos quiere invitar y estos tendrán la opción de finalmente unirse o no a la partida. Un jugador puede participar en varias partidas simultáneamente.

Las jornadas de las que se compone una competición pueden durar días o incluso unas horas. Además, las competiciones deben estar categorizadas por criterios como deporte, populares, recientes... De esta manera se busca registrar las competiciones de forma ordenada. También será posible buscar partidas por criterios como categoría a la que pertenece o popularidad.

En las partidas se deberá poder consultar las apuestas que han realizado el resto de jugadores y ver la clasificación. Para cada amigo que juega se deberá poder ver su perfil, donde se podrá visualizar cuántos puntos tiene y como los ha conseguido.

También se podrá consultar el listado de usuarios registrados, mostrando para cada uno de ellos, sus puntos conseguidos junto a un historial de juego e información acerca de cómo se consiguieron los puntos cronológicamente.

Se deberá poder consultar un listado con todos los juegos que se están jugando y poder filtrar por deporte, competición... Para cada juego se deberá ver la clasificación y al pulsar en cada usuario poder ver los puntos obtenidos cronológicamente solo para ese juego.

Además la aplicación web, módulo que debe proveer todos los servicios web para dar toda la funcionalidad descrita de la aplicación MyBets, tendrá un administrador que se encargara de permitir definir categorías, competiciones, juegos, equipos, partidos y modalidades. Además, el administrador web tendrá la opción de crear partidas publicas en las que todos los jugadores podrán participar. Así mismo, el administrador deberá registrar en el sistema la respuesta correcta al resultado de los partidos.

2.3. Alcance

La parte web del administrador del sistema debe permitir realizar las siguientes acciones:

- Consultar los datos de los juegos, modalidades, usuarios y jornadas
- Definir competiciones y categorías.
- Definir juegos de competiciones.
- Definir las diferentes jornadas de una competición junto a sus plazos y las diferentes opciones posibles en cada jornada.

Por otro lado los servicios web deben permitir realizar las siguientes acciones:

- Permitir al usuario registrarse en el sistema.

- Permitir al usuario identificarse en el sistema.
- Permitir al usuario crear una partida e invitar a diferentes usuarios a unirse.
- Permitir al usuario a aceptar o denegar una invitación.
- Permitir al usuario realizar una apuesta.
- Permitir al usuario consultar su perfil.
- Permitir al usuario consultar sus contactos
- Permitir al usuario consultar las diferentes partidas en las que puede hacer una apuesta.
- Permitir al usuario consultar sus diferentes movimientos en las partidas en las que participa.
- Permitir al usuario cambiar los datos de su perfil

Capítulo 3

Planificación

3.1. Metodología

Para la realización de este proyecto se utilizara la metodología ágil Scrum. Esta metodología se caracteriza por adoptar una estrategia de desarrollo incremental en las que las fases del desarrollo son solapadas. Además basa la calidad de su resultado en el conocimiento tácito de las personas más que en la calidad de los procesos empleados.[3]

Por ello se definirá una pila de producto que deberá realizarse en sprints. En cada sprint se realizará una reunión entre los equipos web, Android e iOS para tomar decisiones de implementación. Los sprints duraran 23 horas y las reuniones periódicas se realizaran al principio de cada sprint.

3.2. Planificación

De las 300 horas de las que están compuestas las prácticas se emplearan 210 horas para la realización de los sprints y sus reuniones que se realizaran cada vez que los 3 equipos hayan acabado un sprint desde la última reunión realizada. En un principio, se realizaría un sprint por semana por lo que como esas 210 horas son aproximadamente 9 semanas se realizaran 9 sprints. Las otras 90 horas serán dedicadas a la formación

3.2.1 Pila del producto inicial (*Initial Product Backlog*)

En la primera reunión se definen una serie de requisitos esenciales que se han de tener en cuenta en todo momento en la evolución del proyecto. La pila del producto está dividida en 2 partes: las funcionalidades de los servicios web que están redactadas bajo el punto de vista de los usuarios que utilizan las aplicaciones de Android y iOS, y por otro lado las necesidades que tiene el portal web bajo el punto de vista del administrador de la aplicación.

A continuación se muestra la pila del producto inicial (Tabla 1):

Id	Nombre	Puntos de historia	Historia de usuario
HU01	Crear partida	3 PH	Yo como usuario quiero poder crear una partida de un juego, con el objetivo de conseguir puntos.
HU02	Elegir la modalidad	1 PH	Yo como usuario quiero poder elegir la modalidad de la competición que estoy creando, para definir la partida.
HU03	Retar amigos	3 PH	Yo como usuario quiero poder elegir los amigos contra los que poder jugar para poder competir contra ellos.
HU04	Realizar una apuesta	3 PH	Yo como usuario quiero poder realizar una apuesta antes de cada jornada con el objetivo de poder ganar puntos o la competición.
HU05	Ganar puntos	3 PH	Yo como usuario quiero poder ganar puntos dependiendo del resultado de la competición para poder clasificarme.
HU06	Modificar perfil	2 PH	Yo como usuario quiero poder modificar mi perfil para actualizar mi información
HU07	Participar en varias partidas	2 PH	Yo como usuario quiero poder participar en varias partidas simultáneamente para poder ganar todas las partidas posibles.
HU08	Consultar competiciones populares	3 PH	Yo como usuario quiero poder ver las competiciones que más se están utilizando para poder crear partidas con mis amigos de esas competiciones.
HU09	Consultar últimos juegos	3 PH	Yo como usuario quiero poder ver los últimos juegos añadidos para poder crear partidas con mis amigos de esos juegos.
HU10	Ver apuestas realizadas dentro de una partida	5 PH	Yo como usuario quiero poder ver las apuestas que han realizado los otros usuarios para poder ver cómo van en la clasificación.
HU11	Consultar juegos por categoría	1 PH	Yo como usuario quiero poder ver un listado de los juegos filtrados por una categoría para crear partidas de los juegos que me interesan.
HU12	Consultar clasificación de una partida	1 PH	Yo como usuario quiero poder ver la clasificación de una partida para poder planear mis siguientes apuestas.
HU13	Visualizar perfil amigo	3 PH	Yo como usuario quiero poder ver el perfil de mis amigos para ver cómo van en las partidas que tenemos en común.

HU14	Consultar próximas jornadas	3 PH	Yo como usuario quiero poder ver las jornadas que se juegan en los próximos días para estar informado sobre las partidas que podría jugar.
HU15	Consultar partidas en curso	2 PH	Yo como usuario quiero poder ver las partidas que tengo en curso y su estado para poder saber qué competiciones tengo pendientes de apostar.
HU16	Consultar muro	5 PH	Yo como usuario quiero poder ver una sección a modo de muro donde poder ver los últimos puntos conseguidos por mí y mis amigos, y las últimas competiciones añadidas para tener una visión general.
HU17	Consultar contactos	5 PH	Yo como usuario quiero poder ver una sección donde poder consultar todos tus contactos para poder invitarlos a jugar.
HU18	Invitar a un amigo	8 PH	Yo como usuario quiero poder invitar amigos a la aplicación para poder jugar contra ellos.
HU19	Notificar fin de la partida	8 PH	Yo como usuario quiero que se me notifique cuando acabe una partida para poder consultar mi resultado.
HU20	Registrarse en el sistema	2 PH	Yo como usuario quiero poder registrarme en el sistema para poder realizar mis apuestas.
HU21	Identificarse en el sistema	1 PH	Yo como usuario quiero poder acceder a mi cuenta para poder realizar apuestas.
HU22	Crear competiciones	3 PH	Yo como administrador quiero poder crear competiciones y sus categorías para que los usuarios puedan crear juegos de esas competiciones.
HU23	Consultar listado de usuarios	2 PH	Yo como administrador quiero poder consultar el listado de usuarios registrados junto a sus puntos conseguidos para estar informado.
HU24	Consultar historial usuario	3 PH	Yo como administrador quiero de un usuario poder ver un historial con sus juegos y los puntos conseguidos cronológicamente para estar informado.
HU25	Introducir resultados	2 PH	Yo como administrador quiero poder introducir los resultados de una jornada para que se actualicen los datos de las competiciones
HU26	Consultar juegos en curso	5 PH	Yo como administrador quiero poder consultar un listado con todos los juegos que se están jugando para así conocer los juegos que se están realizando en mi

			sistema.
HU27	Consultar clasificación partida	3 PH	Yo como administrador quiero ver la clasificación de cada partida para ver quién va ganando.
HU28	Consultar los movimientos de un usuario para una partida	5 PH	Yo como administrador quiero que al pulsar en cada usuario pueda ver los puntos obtenidos cronológicamente solo para ese juego con el objetivo de poder investigar a ese usuario.
Total		91 PH	

Tabla 1. Pila del producto inicial

Tras construir la pila del producto hemos obtenido un total de 91 puntos de historia.

3.2.2. Planificación de iteraciones o *Sprints*

Debido a que los otros equipos de desarrollo necesitan los servicios web para probar sus aplicaciones en un entorno real, las historias de usuario que tengan como usuario al administrador son las que menos prioridad tienen.

Como hay 91 PH por hacer se considera que asignar 10 PH por sprint es una cantidad razonable y adecuada. Así pues, harán falta 9 sprints por lo que al tener que repartir las 210 horas por sprint cada uno de ellos durara 23 horas.

Cabe destacar que el valor de los puntos de historia son orientativos, basados en experiencia personal y en la del equipo de *CuatroOchenta* y que por ello pueden haber historias que a pesar de tener los mismos puntos de historia no tengan exactamente la misma duración estimada. Sin embargo, es necesario señalar que no existirán grandes diferencias entre aquellas del mismo orden.

Considerando todo esto, el reparto de las historias de usuario en los sprints queda de la siguiente forma (Tabla 2).

Sprint	Historia de usuario	Código HU	Tiempo estimado
Sprint 1	Consultar contactos	HU17	14H
	Registrarse en el sistema	HU20	6H
	Identificarse en el sistema	HU21	3H
Sprint 2	Crear partida	HU01	7H
	Elegir la modalidad	HU02	2H
	Retar amigos	HU03	7H
	Consultar últimos	HU09	7H

	juegos		
Sprint 3	Realizar una apuesta	HU04	8H
	Ganar puntos	HU05	8H
	Modificar perfil	HU06	7H
Sprint 4	Participar en varias partidas	HU07	5H
	Consultar competiciones populares	HU08	7H
	Consultar próximas jornadas	HU14	7H
	Consultar partidas en curso	HU15	5H
Sprint 5	Ver apuestas realizadas dentro de una partida	HU10	11.5H
	Consultar juegos por categoría	HU11	2H
	Consultar clasificación de una partida	HU12	2.5H
	Visualizar perfil amigo	HU13	7H
Sprint 6	Consultar muro	HU16	9H
	Invitar a un amigo	HU18	14H
Sprint 7	Notificar fin de la partida	HU19	23H
Sprint 8	Crear competiciones	HU22	7H
	Consultar listado de usuarios	HU23	5H
	Consultar historial usuario	HU24	6H
	Introducir resultados	HU25	5H
Sprint 9	Consultar juegos en curso	HU26	9H
	Consultar clasificación partida	HU27	7H
	Consultar los movimientos de un usuario para una partida	HU28	7H

Tabla 2. Planificación inicial de los sprints

3.3. Estimación de recursos y costes del proyecto

3.3.1. Estimación total del proyecto

En cuanto a los costes del proyecto, considerando el salario del ingeniero informático y otros factores triviales se estima un coste estructural de 21 euros la hora. Por tanto como se va a estar desarrollando la aplicación durante 210 horas a la empresa todo eso le supone un coste de 4410 €.

Además es necesario un servidor para poder mantener la parte web. El servidor utilizado durante el desarrollo del proyecto tiene un coste de 100 euros anuales. Sin embargo si la aplicación tuviese una cantidad de usuarios considerable, del orden de un millón de usuarios, se debería usar el servidor reservado para aplicaciones de ese calibre que cuesta sobre 900 euros anuales.

En cuanto al software, el proyecto se puede desarrollar con software libre por lo que no existen gastos en este apartado. Si sumamos todo eso nos queda que el coste final del proyecto sería de 4510€ considerando en el coste un único año de servidor, 5310 € si la cantidad de peticiones fuese elevada e hiciese falta utilizar el otro.

3.4. Planificación real del proyecto

A continuación se muestran los sprints junto a las fechas previstas y las fechas en las que realmente se realizaron (Tabla 3).

Sprint	Fecha planificada	Fecha real
Sprint 1	4 a 8 de Abril	1 a 7 de Abril
Sprint 2	11 a 14 de Abril	8 a 14 de Abril
Sprint 3	18 a 22 de Abril	15 a 21 de Abril
Sprint 4	25 a 29 de Abril	22 a 28 de Abril
Sprint 5	2 a 6 de Mayo	29 de Abril a 5 de Mayo
Sprint 6	9 a 13 de Mayo	6 a 12 de Mayo
Sprint 7	16 a 20 de Mayo	13 a 19 de Mayo
Sprint 8	23 a 27 de Mayo	20 a 27 de Mayo
Sprint 9	30 de Mayo a 3 de Junio	-

Tabla 3. Fechas de realización de los sprints

Es importante destacar que al final no hubo noveno sprint debido a que no consideré el tiempo que iban a tomar las reuniones de los sprints dentro de los cálculos y cuando llego el 27 de Mayo, termine las 300 horas de estancia.

A continuación se muestra una tabla resumen mostrando cómo se realizo finalmente la implementación durante la estancia (Tabla 4).

Sprint	Historia de usuario	Código HU	Tiempo empleado
Sprint 1	Consultar contactos	HU17	12H
	Registrarse en el sistema	HU20	8H

	Identificarse en el sistema	HU21	3H
Sprint 2	Crear partida y retar amigos ¹	HU01 HU03	11H
	Elegir la modalidad ²	HU02	-
	Consultar últimos juegos	HU09	7H
	Participar en varias partidas*	HU07	5H
Sprint 3	Realizar una apuesta	HU04	16H
	Modificar perfil	HU06	7H
Sprint 4	Consultar competiciones populares	HU08	7H
	Consultar próximas jornadas	HU14	7H
	Consultar partidas en curso	HU15	5H
	Consultar juegos por categoría*	HU11	5H
Sprint 5	Ver apuestas realizadas dentro de una partida	HU10	11.5H
	Consultar clasificación de una partida	HU12	2.5H
	Visualizar perfil amigo	HU13	7H
	Invitar a un amigo*	HU18	2H
Sprint 6	Gestionar categoría	HU29	4.5H
	Consultar muro	HU16	9H
	Consultar listado de usuarios*	HU23	5H
	Gestionar entidades (equipos)	HU30	4.5H
Sprint 7	Crear competiciones e introducir resultados* ³	HU22 HU25	18H
	Definir juego y consultar juegos en curso* ⁴	HU31 HU26	5H

¹ Debido a cambios en los requisitos como consecuencia de una reunión entre el cliente y los equipos web, Android e iOS se decidió que el proceso de retar a los amigos a una partida se realizara en el momento de la creación de la partida por lo que ambas historias de usuario forman parte del mismo procedimiento. Además ahora es necesario almacenar un nombre para cada partida que varía según el usuario.

² Debido a cambios en los requisitos como consecuencia de una reunión entre el cliente y los equipos web, Android e iOS se decidió que la modalidad de una partida viene implícita del juego del que proviene, es decir, es en la creación del propio juego cuando se define esto. Así pues, esta funcionalidad pasa a formar parte de una nueva historia de usuario llamada definir juego que se asigna al séptimo sprint.

³ Debido a cambios en los requisitos como consecuencia de una reunión entre el cliente se decidió que la introducción de los resultados formara parte de la pantalla utilizada para crear una competición. Por ello se decidió combinar estas historias de usuario.

⁴ Estas historias de usuario se combinaron por razones similares a la anterior.

Sprint 8	Ganar puntos	HU05	6.5H
	Consultar clasificación partida*	HU27	6H
	Consultar historial usuario y Consultar los movimientos de un usuario para una partida *	HU24 HU28	10.5H

Tabla 4. Resumen de la implementación de los sprints

Capítulo 4

Análisis y diseño del sistema

Este apartado tiene el objetivo de conseguir un óptimo rendimiento de la aplicación y facilitar posibles extensiones en el futuro .

A continuación se describen datos importantes a tener en cuenta como el estilo arquitectónico de la aplicación o el modelo de la base de datos que se encarga del almacenamiento de los datos de la aplicación.

4.1. Análisis de datos

La aplicación necesita almacenar los datos indispensables para poder cubrir las funcionalidades solicitadas. Por tanto, la información a almacenar por el sistema es:

Requisitos de datos
Código: RD01
Nombre: Usuario
Fuente: Administrador
Versión: 1.2
Requisitos asociados: -
Datos específicos: email, nombre, contraseña, nombre de usuario, teléfono, imagen de perfil
Comentarios:

Requisitos de datos
Código: RD02
Nombre: Categoría
Fuente: Administrador
Versión: 1.2
Requisitos asociados: -
Datos específicos: nombre, color interfaz, icono1, icono2, categoríaPadre
Comentarios:

Requisitos de datos
Código: RD03
Nombre: Competición
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Categoría
Datos específicos: nombre, habilitado, categoría
Comentarios: El campo habilitado permite al administrador ocultar la competición de cara a los usuarios mientras hace cambios en el portal web.

Requisitos de datos
Código: RD04
Nombre: Modalidad
Fuente: Administrador
Versión: 1.2
Requisitos asociados: -
Datos específicos: nombre
Comentarios: En el sistema actual hay soporte para resultado exacto y 1x2

Requisitos de datos
Código: RD05
Nombre: Juego
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Competición, modalidad
Datos específicos: nombre, bote, competición, modalidad
Comentarios:

Requisitos de datos
Código: RD06
Nombre: Tipo
Fuente: Administrador
Versión: 1.2
Requisitos asociados: -
Datos específicos: nombre
Comentarios: En el sistema actual, pública o privada.

Requisitos de datos
Código: RD07
Nombre: Partida
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Juego, Tipo
Datos específicos: juego, tipo
Comentarios:

Requisitos de datos
Código: RD08
Nombre: Jugando
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Usuario, Partida
Datos específicos: nombrePartida, usuario, partida
Comentarios: Representa la relación muchos a muchos con atributos entre usuario y partida. Esto de cara al usuario permite personalizar el nombre de la partida sin afectar al resto de contrincantes.

Requisitos de datos
Código: RD09
Nombre: Amistad
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Usuario
Datos específicos: usuario1, usuario2
Comentarios:

Requisitos de datos
Código: RD10
Nombre: Invitación
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Usuario, Partida
Datos específicos: mensaje, aceptado, usuarioInvitado, usuarioInvita, partida
Comentarios:

Requisitos de datos
Código: RD11
Nombre: Jornada
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Competición
Datos específicos: fechaFin
Comentarios:

Requisitos de datos
Código: RD12
Nombre: Entidad (Equipo)
Fuente: Administrador
Versión: 1.2
Requisitos asociados: -
Datos específicos: nombre, imagen
Comentarios: Se decidió que su nombre sería entidad y no equipo por si en un futuro se ampliase el sistema y se permitiesen apuestas de competiciones no deportivas como pudiera ser el resultado de las elecciones generales o de Eurovisión

Requisitos de datos
Código: RD13
Nombre: Apuesta
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Usuario, Jornada, Partida
Datos específicos: resultado, usuario, jornada, partida
Comentarios: Resultado representa los puntos asociados por el administrador al resultado de una apuesta de un usuario a la jornada de una competición en una partida específica.

Requisitos de datos
Código: RD14
Nombre: PartidoEntidad
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Entidad, Partido
Datos específicos: orden, resultadoReal, partido, entidad
Comentarios: Representa la relación muchos a muchos con atributos entre entidad y partido, es decir, quienes se enfrentan en un partido.

Requisitos de datos
Código: RD15
Nombre: ApuestaPartido
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Partido, Apuesta
Datos específicos: resultado, apuesta, partido
Comentarios: resultado representa el resultado que el usuario predice que sucederá. Este campo solo se utiliza en la modalidad 1x2 y tendrá valor 0,1 o 2

Requisitos de datos
Código: RD16
Nombre: ApuestaPartidoEntidad
Fuente: Administrador
Versión: 1.2
Requisitos asociados: Entidad, ApuestaPartido
Datos específicos: resultado, apuestapartido, entidad
Comentarios: Representa el valor exacto que predice un usuario sobre una entidad en un partido determinado

4.2. Diseño de la arquitectura

El diseño de la arquitectura representa la estructura de los datos y componentes de la aplicación necesarios para la implementación del sistema informático así como sus propiedades e interrelación.

En un proyecto software como este, este proceso es importante por tres razones fundamentales:

- En primer lugar las representaciones de la arquitectura **facilitan la comunicación** entre todos los involucrados en su desarrollo. Si se considera que en este proyecto están trabajando los equipos de desarrollo web, Android e iOS obtenemos que esto puede reducir drásticamente la cantidad de malentendidos que se pueden dar durante el proceso de implementación.
- En segundo lugar, tal y como se ha hecho alusión antes, obliga a tomar **decisiones tempranas** que tienen un impacto importante en el desarrollo del sistema.
- En tercer lugar, la visión global ofrecida **facilita la comprensión** de la estructuración del sistema y de cómo trabajan los componentes de forma conjunta.

4.2.1. Arquitectura del sistema

Se van a realizar servicios REST que van a dar servicio a las aplicaciones iOS y Android (Figura 1). Esto significa que las peticiones son stateless, es decir, no tienen estado [4]. Es por eso que en todos los JSON de los servicios web que se mandan como petición siempre se manda la información del usuario (para más información consúltase el anexo A).

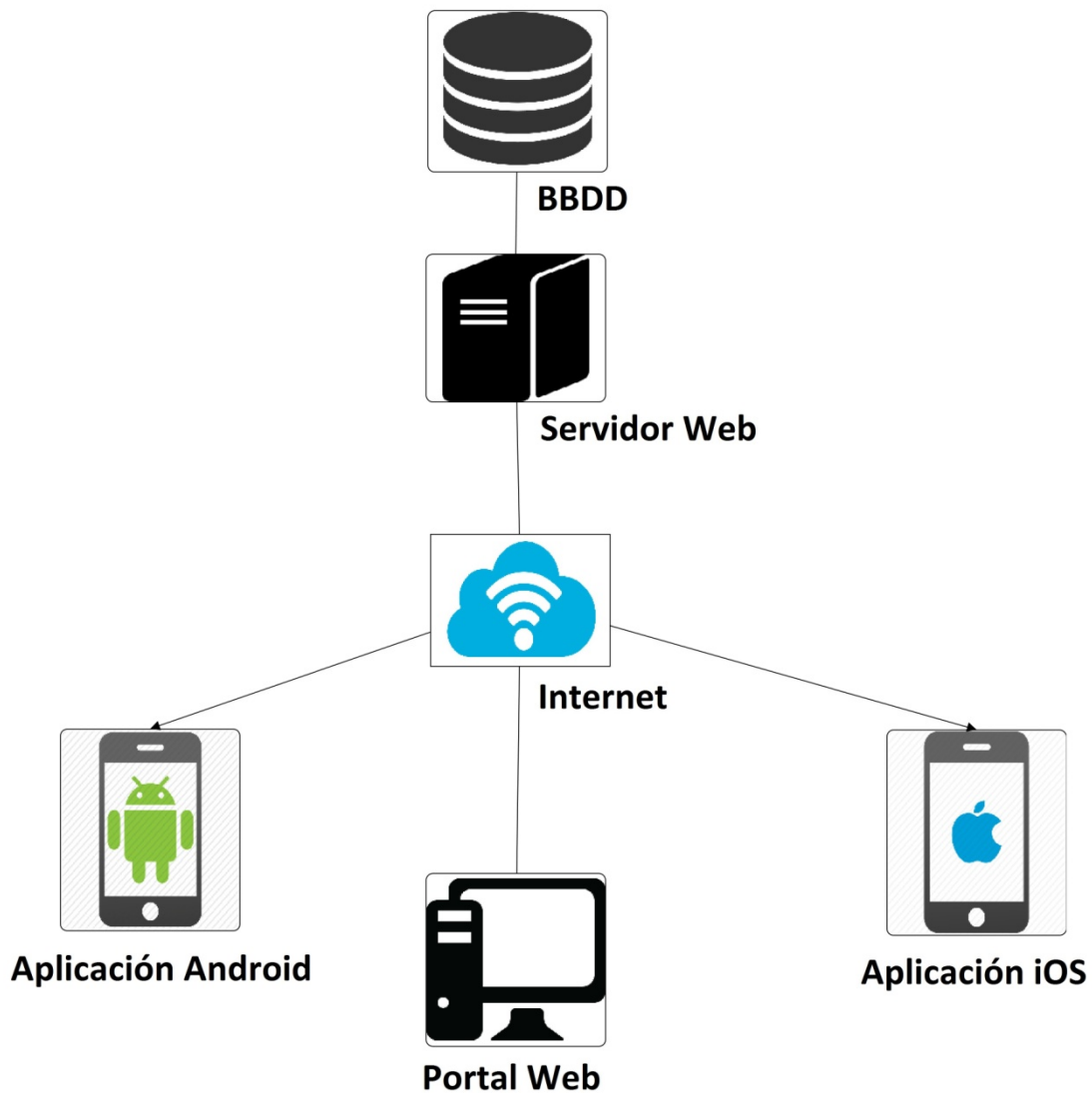


Figura 1. Arquitectura del sistema

El estilo de este sistema es una **combinación** de varios estilos arquitectónicos. Este hecho es muy común en las aplicaciones orientadas a dar soporte a aplicaciones móviles.

Los estilos que se funden para dar lugar a la arquitectura de este sistema son el estilo arquitectónico **centrado en los datos**, el **orientado a objetos** y la arquitectura **cliente/servidor**, que es necesaria para que los servicios sean REST [5].

Las arquitecturas centradas en los datos están construidas alrededor de un **almacén de datos** donde los clientes operan de forma independiente permitiendo la **comunicación** a través de él. En este caso, la parte del servidor se encarga de gestionar ese acceso al almacén de datos, que es la base de datos de la que se habló en el apartado anterior, y los jugadores se comunican entre ellos ya sea apostando o uniéndose a partidas a través de este.

Las arquitecturas orientadas a objetos **encapsulan** los datos y las operaciones que se le aplican para **manipularlos**. La comunicación se realiza mediante paso de mensajes.

Por tanto, si tenemos estos 2 estilos en cuenta, obtenemos un sistema donde los componentes están contruidos alrededor de un almacén de datos permitiendo la comunicación entre todos los componentes conectados a él y estos, a su vez, encapsularan todos los datos y aplicaran las operaciones necesarias para manipularlos.

Finalmente queda la arquitectura cliente/servidor. En este caso el portal web y las aplicaciones Android e iOS cumplen el rol del cliente. Estos se comunicaran con el servidor y realizaran **peticiones** con la finalidad de ofrecer y obtener información. Todos ellos (que se pueden considerar como subsistemas) forman parte del sistema completo ya que interactúan entre ellos para lograr un **objetivo común**.

A continuación se describen los subsistemas y las relaciones entre ellos:

- **BackOffice:** En este subsistema se alojan la base de datos del sistema y los servicios web a través de los cuales las aplicaciones móviles pueden acceder a esta información. Como el intercambio de mensajes se hace en formato JSON, nos permite realizar una comunicación entre varios elementos del sistema sin importar al subsistema al que pertenezcan.
- **Portal web:** Website accesible para el administrador. En ella se muestra la información necesaria que da vida a la aplicación. Este componente se comunica con el servidor (BackOffice) a través del protocolo HTTP.
- **Aplicaciones Android e iOS:** Este subsistema es el medio que utilizara la mayor cantidad de usuarios de la aplicación para interactuar con el sistema. La aplicación se conecta con el BackOffice a través de los

servicios web REST y la información es presentada de forma amigable para el usuario a través de la aplicación.

4.2.2. Diseño de clases

El diseño de clases representa las entidades que formaran parte del sistema, es decir, representa la información que el servidor debe almacenar y que las aplicaciones clientes necesitaran que les provean para poder ofrecer la funcionalidad esperada.

Por el tipo de información y sus relaciones el modelo de datos relacional es un modelo que se ajusta a nuestro sistema, así que se utilizara este.

Para poder facilitar la coordinación entre la base de datos y las operaciones de los servicios web a implementar se utilizara un ORM (*Object-Relational Mapping*) que proporciona una capa de persistencia para objetos. Esto se utiliza para realizar las búsquedas en la base de datos ya que abstrae el diseño físico de la base de datos y únicamente es necesario conocer los atributos de las clases que encapsularan la información deseada, permitiéndonos trabajar con toda esta información como si estuviese guardada totalmente en memoria.

A continuación se presenta el diseño final de la base de datos utilizada en la aplicación (Figura 2).

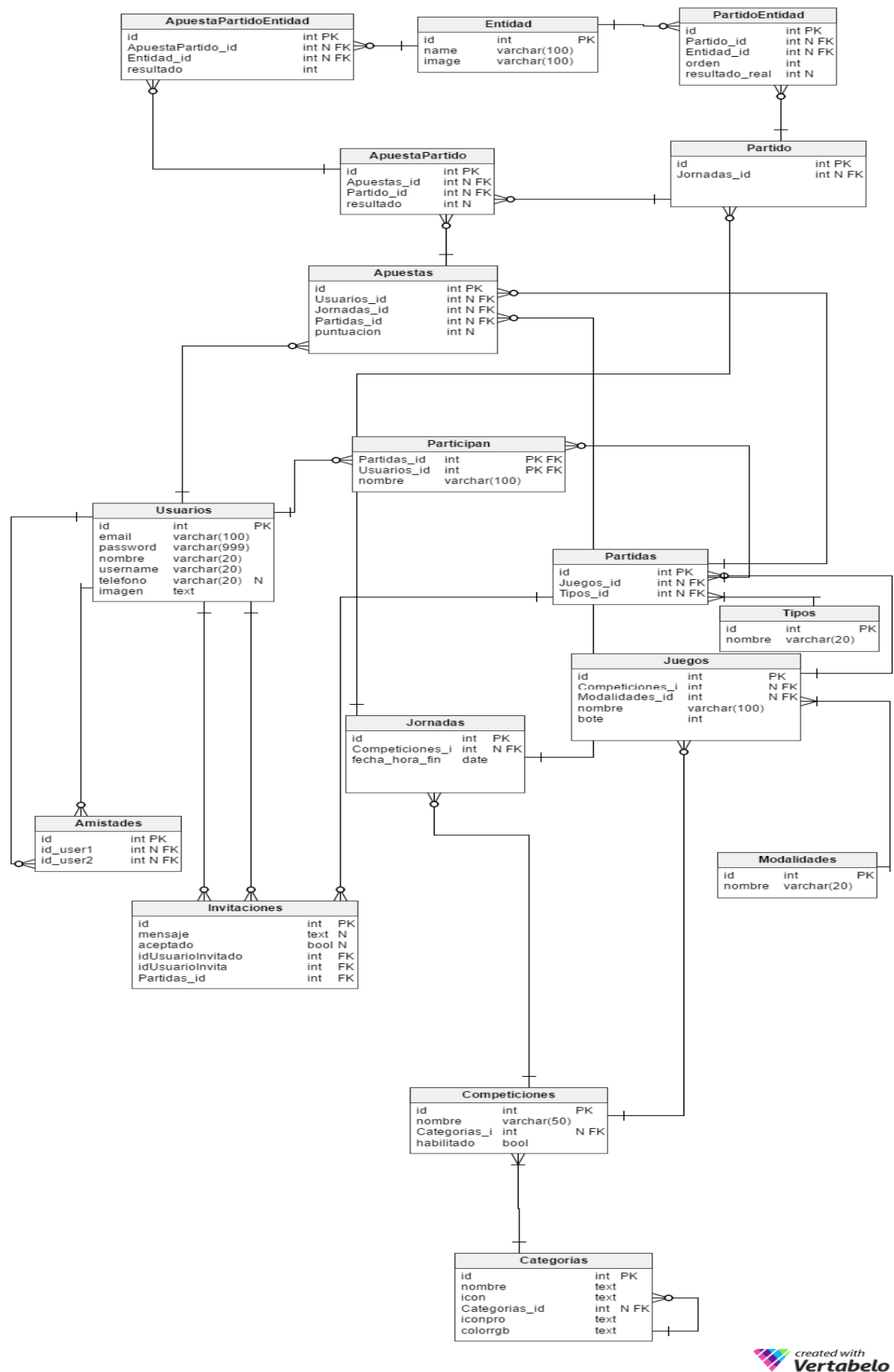


Figura 2. Modelo de la base de datos

4.3. Diseño de la interfaz

En cuanto al diseño de la interfaz del portal web del administrador de este proyecto, debido a la gran cantidad de información a gestionar, es importante definir primero una jerarquía de pantallas, ya que ayuda a comprender que se busca alcanzar y nos permite tomar **decisiones tempranas**. Por ejemplo, si se diese el caso de que existiera un gran número de páginas de un mismo nivel podríamos decidir que nuestra interfaz gráfica necesitará un menú para poder navegar entre ellas de forma simple.

A continuación se muestra la jerarquía de páginas del portal web (Figura 3):



Figura 3. Mapa del sitio

4.4. Diseño de la página web (wireframe)

Teniendo en cuenta la información del apartado anterior, para el diseño de la interfaz gráfica se ha decidido utilizar un template gratuito, SB Admin de Start Bootstrap (Figura 4), intentando cumplir las 8 reglas de oro en diseño de interfaces en la mayor medida posible [6]

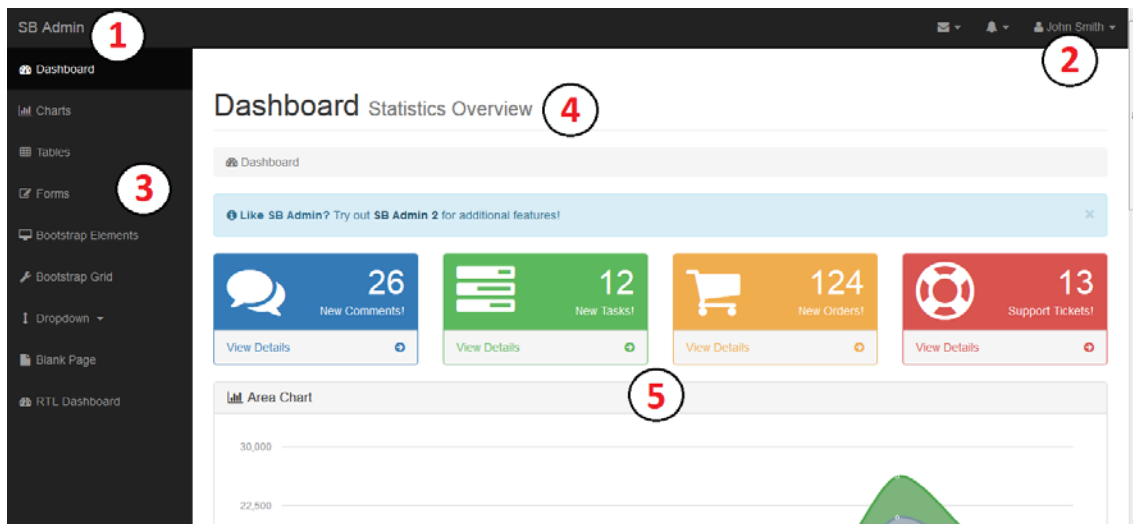


Figura 4. Template utilizado para la interfaz gráfica

En la zona 1 esta la cabecera que llevara a la página principal del portal web, disponible en todo momento. Además se utilizará el logotipo de la aplicación que usan las aplicaciones iOS y Android para que visualmente el usuario, en este caso el administrador, tenga claro en todo momento que el portal web forma parte de la aplicación.

Por otro lado en la zona 2 se desecharan los iconos del sobre y la campana de la esquina superior derecha debido a la ausencia de funcionalidades por implementar que puedan beneficiarse de ellos. En el icono de la persona aparecerá el nombre del usuario identificado y se mostrara un desplegable con la opción de cerrar sesión.

A la izquierda, en la zona 3, encontramos un menú que se utilizara para navegar entre las páginas del tercer nivel de la figura 2. De esta manera se le otorga al usuario gran libertad para moverse entre los distintos elementos del sistema que hace falta gestionar, dándole un mayor control de la misma.

Además el texto de la esquina superior izquierda será sustituido por el logotipo de la aplicación, presente también en las pantallas de las aplicaciones Android e iOS.

En la zona 4 se encuentra el título de la sección en la que se encuentra el usuario, con el objetivo de reducir al máximo posible su carga de memoria y facilitar en todo momento en que parte del portal web se encuentra.

Finalmente en la zona 5 representa el contenido de la sección. Para la página principal se utilizaran las cajas de colores para mostrar información del sistema como la cantidad de usuarios registrados, el número de juegos registrados en el sistema o el número de categorías disponibles.

Capítulo 5

Implementación

5.1. Tecnologías y herramientas utilizadas

Las tecnologías utilizadas en el proyecto de forma general son las siguientes:

- **Apache:** Núcleo del servidor de ficheros distribuido que emplea el protocolo HTTP para comunicarse con los clientes que piden acceso a ciertos archivos o directorios.
- **HTTP:** Medio de comunicación entre el servidor y cliente
- **Symfony:** Framework que se encarga de la gestión de contenidos. En este proyecto se utiliza la versión **2.7** [7].

En cuanto a las herramientas se han utilizado las siguientes:

- **Phpstorm:** Entorno de programación web orientado a PHP.
- **Wamp:** Plataforma de desarrollo web empleada para mantener una réplica del sistema ya que permite fácilmente configurar e instalar Apache, MySQL y PHP y poder realizar pruebas en local.

5.1.1. Implementación del portal web

Para la implementación del portal web se han utilizado además las siguientes tecnologías:

- **HTML:** Lenguaje para la elaboración de páginas web. En este proyecto se utiliza la versión **5**.
- **Bootstrap:** Estilos CSS para la página web.

5.1.2. Implementación de la base de datos

Para la implementación de la base de datos de la aplicación se han utilizado las siguientes tecnologías:

- **MySQL:** Gestor de bases de datos para la persistencia de los datos de la aplicación

En cuanto a las herramientas se han utilizado las siguientes:

- **HeidiSQL:** Aplicación utilizada para gestionar las bases de datos del servidor y la de pruebas.

5.1.3. Implementación de los servicios web

Para los servicios web se ha utilizado lo siguiente:

- **JSON:** Formato común de los datos de comunicación entre el servidor y el cliente

En cuanto a las herramientas se han utilizado las siguientes:

- **Advanced REST Client:** Plugin de **Google Chrome** utilizado para poder probar los servicios web mediante el envío y recepción de JSON.

Capítulo 6

Desarrollo proyecto

En este apartado se describe con más detalle el proceso de implementación del proyecto mediante metodología ágil.

6.1 Sprint 1

A continuación se muestra la pila del sprint 1 (Tabla 5. Planificación del sprint 1), formada por 3 historias de usuario descompuestas en tareas que nos proporcionarán una estimación aproximada del tiempo que nos llevará implementarlo.

Código HU	Historia de usuario	Tareas	Código tarea
HU21	Yo como jugador quiero poder acceder a mi cuenta para poder realizar apuestas. (1PH)	Crear entidad usuario con todos sus datos.	T001
		Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T002
		Crear un servicio REST que permita validar el acceso a la cuenta del usuario.	T003
HU20	Yo como jugador quiero poder registrarme en el sistema para poder realizar mis apuestas. (5PH)⁵	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T004
		Crear servicio REST que permita la creación de nuevos usuarios en el sistema	T005
		Investigar e implementar de forma conjunta un sistema para codificar una imagen png en base64 para poder registrar una imagen de perfil para el usuario.	T006

⁵ El valor en puntos de historia ha cambiado respecto a la planificación original por un cambio de requisitos. En un principio no estaba contemplada la posibilidad de subir un fichero png como imagen de usuario lo que ha conllevado a una investigación por parte de los 3 equipos para poder cumplir con este nuevo requisito.

HU17	Yo como jugador quiero poder ver una sección donde poder consultar todos tus contactos para poder invitarlos a jugar. (5 PH)	Creación de entidad invitación con todos sus datos.	T007
		Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T008
		Creación de un servicio REST para permitir consultar todos los amigos disponibles	T009
		Añadir lógica para obtener los amigos de un usuario para impedir realizar acciones restringidas sobre usuarios que no sean amigos y reutilizarlo en otras funcionalidades.	T010

Tabla 5. Planificación del sprint 1

6.2. Sprint 2

A continuación se muestra la pila del sprint 2 (Tabla 6. Planificación del sprint 2), formada originalmente por 4 historias de usuario descompuestas en tareas que nos proporcionarán una estimación aproximada del tiempo que nos llevará implementarlo. Sin embargo, debido a cambios en los requisitos del usuario algunas historias de usuario desaparecieron para pasar a formar parte de otras y se añadieron algunas nuevas.

Código HU	Historia de usuario	Tareas	Código tarea
HU02	Yo como jugador quiero poder elegir la modalidad de la competición que estoy creando, para definir la partida. (3 PH)⁶	-	-
HU01 HU03	Yo como jugador	Creación de entidad	T011

⁶ Debido a cambios en los requisitos como consecuencia de una reunión entre el cliente y los equipos web, Android e iOS se decidió que la modalidad de una partida viene implícita del juego del que proviene, es decir, es en la creación del propio juego cuando se define esto. Así pues, esta funcionalidad pasa a formar parte de una nueva historia de usuario llamada definir juego que se asigna al octavo sprint.

	quiero poder crear una partida de un juego, con el objetivo de conseguir puntos. Yo como jugador quiero poder elegir los amigos contra los que poder jugar para poder competir contra ellos. (8PH)⁷	competición con todos sus datos.	
		Creación de entidad juego con todos sus datos.	T012
		Creación de entidad modalidad con todos sus datos.	T013
		Creación de entidad partida con todos sus datos.	T014
		Creación de entidad participan con todos sus datos.	T015
		Creación de entidad tipo con todos sus datos.	T016
		Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T017
		Creación de un servicio REST para permitir la creación de una partida de un juego así como invitar a otros usuarios a participar.	T018
HU07	Yo como jugador quiero poder participar en varias partidas simultáneamente para poder ganar todas las partidas posibles. (2 PH)⁸	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T019
		Creación de un servicio REST para permitir la aceptación de una invitación para formar parte de la partida.	T020
		Creación de la	T021

⁷ Debido a cambios en los requisitos como consecuencia de una reunión entre el cliente y los equipos web, Android e iOS se decidió que el proceso de retar a los amigos a una partida se realizara en el momento de la creación de la partida por lo que ambas historias de usuario forman parte del mismo procedimiento. Además ahora es necesario almacenar un nombre para cada partida que varía según el usuario.

⁸ Originalmente esta historia de usuario estaba asignada al cuarto sprint pero debido a su conexión con la historia anterior se decidió implementarla finalmente en este para que la funcionalidad de las invitaciones estuviera ya completa.

		lógica necesaria para aceptar partidas y para filtrar aquellas invitaciones pendientes por aceptar o rechazar.	
HU09	Yo como jugador quiero poder ver los últimos juegos añadidos para poder crear partidas con mis amigos de esos juegos. (3 PH)	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T022
		Creación de un servicio REST para permitir la consulta de últimos juegos disponibles.	T023

Tabla 6. Planificación del sprint 2

6.3. Sprint 3

A continuación se muestra la pila del sprint 3 (Tabla 7. Planificación del sprint 3), originalmente formada por 3 historias de usuario descompuestas en tareas que nos proporcionarán una estimación aproximada del tiempo que nos llevará implementarlo. Sin embargo, debido a cambios en los requisitos del usuario que aparecieron en la reunión del sprint, una historia de usuario se atrasó y otra de un sprint posterior fue adelantada a este.

Código HU	Historia de usuario	Tareas	Código tarea
HU04	Yo como jugador quiero poder realizar una apuesta antes de cada jornada con el objetivo de poder ganar puntos o la competición. (8 PH)⁹	Creación de entidad jornada con todos sus datos.	T024
		Creación de entidad partido con todos sus datos.	T025
		Creación de entidad con todos los datos que definen un participante de un partido	T026
		Creación de entidad apuesta con todos sus datos.	T027
		Creación de entidad apuestaPartido con todos sus datos.	T028
		Creación de entidad ApuestaPartidoEntidad con todos sus datos.	T029

⁹ Nuevamente, los puntos de historia fueron alterados, esta vez debido a malentendidos entre los componentes de los tres equipos y el cliente que conllevó a que inicialmente se subestimara el coste de esta historia de usuario.

		Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T030
		Creación de la lógica necesaria para registrar correctamente una apuesta según el tipo de partida sobre la que se esté jugando	T031
		Creación de un servicio REST para permitir la realización de la apuesta en una partida	T032
HU05	Yo como jugador quiero poder ganar puntos dependiendo del resultado de la competición para poder clasificarme. (3 PH)¹⁰	-	-
HU06	Yo como jugador quiero poder modificar mi perfil para actualizar mi información. (2 PH)	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T033
		Creación de un servicio REST para permitir la realización de la modificación del perfil de un usuario	T034
		Creación de la lógica necesaria para garantizar que un jugador únicamente puede modificar su propio perfil y no de una tercera persona	T035

Tabla 7. Planificación del sprint 3

¹⁰ Esta historia de usuario se pospuso a un sprint posterior debido a la reunión del sprint. Se decidió que los puntos se asignarían a los jugadores automáticamente cuando el administrador lo solicitase desde el portal web.

6.4. Sprint 4

A continuación se muestra la pila del sprint 4 (Tabla 8. Planificación del sprint 4), originalmente formada por 4 historias de usuario descompuestas en tareas que nos proporcionarán una estimación aproximada del volumen de trabajo que nos llevará implementarlo. Una historia fue implementada en el anterior sprint por lo que se tomó una historia de usuario del sprint posterior.

Código HU	Historia de usuario	Tareas	Código tarea
HU08	Yo como usuario quiero poder ver las competiciones que más se están utilizando para poder crear partidas con mis amigos de esas competiciones. (3 PH)	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T036
		Creación de un servicio REST para permitir la obtención de los datos de las competiciones más populares	T037
HU14	Yo como usuario quiero poder ver las jornadas que se juegan en los próximos días para estar informado sobre las partidas que podría jugar. (3 PH)	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T038
		Creación de un servicio REST para permitir la obtención de los datos de las partidas en las que se puede realizar una apuesta.	T039
HU15	Yo como usuario quiero poder ver las partidas que tengo en curso y su estado para poder saber qué competiciones tengo pendientes de apostar. (5 PH)¹¹	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS	T040
		Creación de un servicio REST para permitir la obtención de los datos de las partidas cuyo plazo de votación aun no haya terminado, junto a su estado.	T041

¹¹ Los puntos de historia aquí fueron incrementados debido a que se fragmentó la historia de usuario en dos servicios web y que en la reunión del sprint se decidió que cuando ha terminado el plazo de una partida y no quedan más jornadas por realizar hasta que no se le asignen los puntos esta no debe aparecer.

		Creación de un servicio REST para permitir la obtención de los datos de las partidas cuyo plazo de votación haya terminado ya y el administrador haya asignado ya los puntos.	T042
HU11	Yo como usuario quiero poder ver un listado de los juegos filtrados por una categoría para crear partidas de los juegos que me interesan. (2 PH)¹²	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS.	T043
		Creación de entidad categoría con todos sus datos.	T044
		Creación de un servicio REST para permitir la obtención de los juegos filtrados por una categoría.	T045

Tabla 8. Planificación del sprint 4

6.5. Sprint 5

A continuación se muestra la pila del sprint 5 (Tabla 9. Planificación del sprint 5), originalmente formada por 4 historias de usuario. Como una historia fue adelantada se decidió adelantar una historia del siguiente sprint.

Código HU	Historia de usuario	Tareas	Código tarea
HU10	Yo como usuario quiero poder ver las apuestas que han realizado los otros usuarios para poder ver cómo van en la clasificación. (5 PH)	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS.	T046
		Creación de un servicio REST para permitir la consulta de las apuestas	T047

¹² Aquí cabe destacar una modificación realizada durante la reunión del sprint y es la aparición de "subcategorías", es decir, categorías que dependen de otras en las búsquedas, por ejemplo el caso fútbol, tenis y deportes. Las 3 son categorías del sistema (por lo que la HU en si no cambia) pero tenis y fútbol son deportes y por tanto para los detalles visuales de la interfaz gráfica de las aplicaciones se debe proporcionar la información correspondiente a deportes. Por ello los puntos de historia aumentaron.

		realizadas por jugadores en una partida a la que el usuario pertenece.	
HU12	Yo como usuario quiero poder ver la clasificación de una partida para poder planear mis siguientes apuestas. (1 PH)	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS.	T048
		Creación de un servicio REST para permitir la consulta de la clasificación de jugadores de una partida en la que el usuario este participando.	T049
HU13	Yo como usuario quiero poder ver el perfil de mis amigos para ver cómo van en las partidas que tenemos en común. (3 PH)	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS.	T050
		Creación de un servicio REST para permitir la consulta de los resultados de las apuestas de partidas en las que este participando el usuario	T051
HU18	Yo como usuario quiero poder invitar amigos a la aplicación para poder jugar contra ellos. (1 PH)¹³	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS.	T052
		Creación de un servicio REST para permitir forjar una amistad entre dos usuarios.	T053

Tabla 9. Planificación del sprint 5

¹³ Los puntos de historia bajaron drásticamente debido a que para implementar la funcionalidad se reutilizo parte de la HU consultar contactos que se realizó en un sprint anterior.

6.6. Sprint 6

A continuación se muestra la pila del sprint 6 (Tabla 10. Planificación del sprint 6), originalmente formada por dos historias de usuario. Sin embargo, las asignaciones de este sprint cambiaron notablemente debido, por una parte, al adelanto de una historia de usuario y por otra a la aparición de nuevas historias de usuario que surgieron en la reunión del sprint.

Código HU	Historia de usuario	Tareas	Código tarea
HU29	Yo como administrador quiero poder gestionar las categorías que hay en el sistema para poder crear borrar y modificarlas. (5 PH)	Creación de la clase CategoryType que representa la entidad Categoría para poder crear el formulario	T054
		Creación de las plantillas Twig para permitir consultar, listar crear y editar categorías.	T055
		Creación de la lógica necesaria en el controlador para rellenar las plantillas, gestionar las peticiones de creación, actualización y borrado y dibujar el formulario a partir de su clase type además de permitir la subida de imágenes y selección del color de la categoría.	T056
HU16	Yo como usuario quiero poder ver una sección a modo de muro donde poder ver los últimos puntos conseguidos por mí y mis amigos, y las últimas competiciones añadidas para tener una visión	Definir los JSON de envío y respuesta entre los equipos web, Android e iOS.	T057
		Creación de un servicio REST para permitir consultar las últimas asignaciones de puntos para el usuario y para sus amigos	T058

	general. (5 PH)		
HU23	Yo como administrador quiero poder consultar el listado de usuarios registrados junto a sus puntos conseguidos para estar informado. (2 PH)	Creación de la plantilla Twig necesaria para poder listar los usuarios del sistema junto a sus puntos.	T059
		Creación de la lógica necesaria en el controlador para rellenar la plantilla	T060
HU30	Yo como administrador quiero poder gestionar aquellos entes que se enfrentan en los partidos de mi aplicación. (2 PH)	Creación de la clase EntidadType que representa la entidad Entidad (equipo) para poder crear el formulario	T061
		Creación de las plantillas Twig para permitir consultar, listar crear y editarlos.	T062
		Creación de la lógica necesaria en el controlador para rellenar las plantillas y dibujar el formulario además de permitir la subida de imágenes de los iconos que los representan.	T063

Tabla 10. Planificación del sprint 6

6.7. Sprint 7

A continuación se muestra la pila del sprint 7 (Tabla 11. Planificación del sprint 7), que originalmente era el sprint 8 pero tras la reunión del sprint se decidió atrasar el viejo 7 al final del todo ya que las demás funcionalidades se consideraban mas prioritarias. Esto ocasiono un reajuste en la planificación haciendo que algunas historias de usuario intercambiaran posiciones con el siguiente sprint.

Código HU	Historia de usuario	Tareas	Código tarea
HU22 HU25	Yo como administrador quiero poder crear competiciones y sus categorías para que los usuarios puedan crear juegos de esas competiciones. Yo como administrador quiero poder introducir los resultados de una jornada para que se actualicen los datos de las competiciones. (8 PH)	Creación de la clase CompetitionType que representa la entidad Competición para poder crear el formulario	T064
		Creación de la clase JornadaType que representa la entidad Jornada para poder crear el formulario	T065
		Creación de la clase PartidoType que representa la entidad Partido para poder crear el formulario	T066
		Creación de la clase PartidoEntidadType que representa la entidad PartidoEntidad para poder crear el formulario	T067
		Creación de las plantillas Twig para permitir consultar, listar crear y editar competiciones.	T068
		Creación de la lógica necesaria en el controlador para rellenar las plantillas, gestionar las peticiones de creación, actualización y borrado y dibujar el formulario de competición a partir de su clase type	T069

		Creación de las plantillas Twig para permitir consultar, listar crear y editar jornadas.	T070
		Creación de un script en javascript para permitir añadir y quitar de forma dinámica partidos al formulario para así permitir asignar a una jornada tantos partidos como se desee	T071
		Creación de un script en javascript para la página de editar jornada para presentar los partidos que ya estaban asignados en el sistema con anterioridad de la misma forma que los que se crean dinámicamente	T072
		Creación de la lógica necesaria para permitir añadir actualizar y borrar jornadas y a su vez de los partidos junto a sus contrincantes y conseguir persistirlos de forma correcta consiguiendo guardar de forma correcta el orden de presentación de estos, es decir, que se guarde quien es el equipo local y cual el visitante por su posición en la pantalla.	T073
HU31 HU26	Yo como administrador quiero poder gestionar los juegos que se juegan en el sistema asignándoles la modalidad que	Creación de la clase GameType que representa la entidad Juego para poder crear el formulario	T074
		Creación de la clase ModalityType que representa la entidad Modalidad para	T075

	desee para que el resto de usuarios puedan jugarlos y decidiendo yo si el juego está disponible para el resto o no.	poder crear el formulario	
	Yo como administrador quiero poder consultar un listado con todos los juegos que se están jugando para así conocer los juegos que se están realizando en mi sistema. (2 PH)	Creación de la clase PartidaType que representa la entidad Partida para poder crear el formulario	T076
		Creación de las plantillas Twig para permitir consultar, listar crear y editar juegos.	T077
		Creación de las plantillas Twig para permitir consultar, listar crear y editar partidas disponibles para todos los usuarios.	T078
		Creación de la lógica necesaria en el controlador para rellenar las plantillas, gestionar las peticiones de creación, actualización y borrado y dibujar el formulario de juego a partir de su clase type	T079
		Creación de la lógica necesaria en el controlador para rellenar las plantillas, gestionar las peticiones de creación, actualización y borrado y dibujar el formulario de partida a partir de su clase type	T080

Tabla 11. Planificación del sprint 7

6.8. Sprint 8

A continuación se muestra la pila del sprint 8 (Tabla 12. Planificación del sprint 8), que originalmente era el sprint 9. En este sprint se terminaron de implementar el resto de funcionalidades menos la de notificar fin de la partida.

Código HU	Historia de usuario	Tareas	Código tarea
HU05	Yo como usuario quiero poder ganar puntos dependiendo del resultado de la competición para poder clasificarme. (3 PH)	Creación de la lógica necesaria para el cálculo del reparto de puntos según la modalidad del juego que se está jugando y la asignación de estos a los jugadores	T081
HU27	Yo como administrador quiero ver la clasificación de cada partida para ver quién va ganando. (3 PH)	Reformar la plantilla Twig utilizada para consultar una partida para mostrar los datos de la clasificación	T082
		Creación de la lógica necesaria para recopilar toda la información de la clasificación y mostrarla	T083
HU24 HU28	Yo como administrador quiero de un usuario poder ver un historial con sus juegos y los puntos conseguidos cronológicamente para estar informado. Yo como administrador quiero que al pulsar en cada usuario pueda ver los puntos obtenidos cronológicamente solo para ese juego con el objetivo de poder	Creación de la plantilla Twig necesaria para consultar un usuario y presentar de forma ordenada para cada partida jugada la fecha en la que se apostó y los puntos obtenidos por ello	T084
		Creación de la lógica necesaria para recopilar toda la información del usuario junto a sus apuestas	T085

	investigar a ese usuario. (5 PH)		
--	---	--	--

Tabla 12. Planificación del sprint 8

Capítulo 7

Conclusiones

7.1. Resumen

Como resumen del proyecto se puede concluir con que la gestión del proyecto es tan importante como las herramientas y tecnologías utilizadas para la implementación. El uso de metodología ágil como es Scrum facilita mucho las cosas pero no es infalible para prevenir imprevistos.

Otro aspecto a destacar es el pensamiento general que se suele tener con relacionar la metodología ágil con la ausencia de documentación del sistema. Esto es totalmente falso, y es que en un proyecto como este, donde la colaboración con los equipos Android e iOS tiene tanto peso en el resultado final, si no se hubiesen documentado todas las especificaciones técnicas de las peticiones de los servicios web o de las funcionalidades habría sido imposible alcanzar la mitad de las funcionalidades de las que dispone el resultado final.

Por otro lado, cabe destacar la importancia de las reuniones de los sprints y con el cliente, que previnieron que muchos errores que podrían haber acabado en el fracaso de toda la planificación realizada y la reimplementación de más de medio proyecto fuesen cometidos.

7.2. Objetivos logrados

Tras finalizar el proyecto se ha obtenido una primera versión completamente funcional del sistema que fue descrito y analizado a lo largo de este documento, proporcionando un portal web para el administrador con una interfaz agradable y amigable que permite gestionar los contenidos de MyBets de forma satisfactoria y unos servicios web, robustos, completamente funcionales que utilizan JSON para el envío y respuesta de las peticiones.

7.3. Valoración personal

A nivel personal este ha sido el proyecto más gratificante y ambicioso en el que he trabajado hasta la fecha. Al comienzo de todo esto, yo solo tenía nociones básicas de HTML y Javascript y nulos conocimientos de PHP, que es un

lenguaje de programación que no se imparte en mi itinerario. En solo 2 meses, aprendí a programar en PHP desde cero y me familiarice con Symfony que es un framework enorme con una estructura muy modular y muy interesante con un montón de posibilidades gracias a la existencia de los bundles de terceros que permiten crear cosas verdaderamente espectaculares y asombrosas.

Además me sirvió para familiarizarme con conceptos como son JSON, REST, los cuales ya conocía pero nunca había tenido la ocasión de trabajar con ellos hasta ahora.

Considerando que todos los proyectos grandes en los que he trabajado han sido en java, lenguaje que se nos inculca desde el primer curso del grado, ha sido una magnífica ocasión para superarme a mí mismo y para aplicar de forma práctica todos los conocimientos que he obtenido en el transcurso de estos 4 años de carrera así como a desenvolverme en un entorno profesional.

Por último, me gustaría agradecer profundamente al equipo de *CuatroOchenta* por lo bien que me recibieron y me han tratado hasta el último momento así como a mis compañeros de proyecto cuya actitud colaborativa ha podido hacer posible que este proyecto tuviese éxito.

Bibliografía

- [1] Franbarquilla. Estudio sobre el estado de Internet y las redes sociales en 2016. <http://franbarquilla.com/estudio-sobre-el-estado-de-Internet-y-las-redes-sociales-en-2016>. [Consulta: 27 de Junio de 2016].
- [2] Wikipedia. Web Service. https://en.wikipedia.org/wiki/Web_service. Web service. [Consulta: 21 de Junio de 2016].
- [3] Juan Palacio. Gestión de proyectos Scrum Manager, versión 2.5. http://www.scrummanager.net/files/sm_proyecto.pdf. [Consulta: 18 de Julio de 2016]
- [4] Wikipedia. Representational state transfer. https://en.wikipedia.org/wiki/Representational_state_transfer. [Consulta: 21 de Junio de 2016].
- [5] EI1039 Diseño del software, Tema 2 Diseño de la arquitectura. [Consulta: 30 de Junio del 2016].
- [6] EI1027, Diseño e implementación de sistemas de la información, Diseño de interfaces de usuario, usabilidad, guías, principios y teorías. [Consulta: 30 de Junio del 2016].
- [7] The Symfony Book 2.7, <http://symfony.com/doc/2.7/book/index.html>. [Consulta: 18 de Marzo del 2016].

Anexo A

Estudio detallado de servicios web desarrollados y JSON de envío y respuesta

A.1. Definición

En esta sección se presentan todos los servicios web implementados en este proyecto junto a sus url y los JSON de petición y respuesta establecidos.

El formato para fechas es “dd/MM/yyyy HH:mm ” como por ejemplo “12/03/2045 17:58”.

BaseURL = mybetstest.cuatroochenta.com

Login

URL: http://[BASE_URL]/services/mybets_login

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}
```

Respuesta:

```
{
  "data": {
    "id": "id:Integer",
    "email": "email:String",
    "name": "nombre:String",
    "username": "nick:String",
    "phone": "teléfono:String",
  }
}
```

```
},  
  "code": "SUCCESS (0) || ERROR (1)",  
  "message": "STRING VACÍO || MENSAJE DE ERROR"  
}
```

Registro

URL: http://[BASE_URL]/services/mybets_register

Método: POST

Parámetros:

```
{  
  "request":  
    {  
      "email": "email:String",  
      "password": "contraseña:String(md5)",  
      "name": "nombre:String",  
      "username": "nick:String",  
      "phone": "telefono:String",  
      "image" : "imagen:String(Base64)"  
    }  
}
```

Respuesta:

```
{  
  "data": {  
    "id": "id:Int",  
    "email": "email:String",  
    "name": "nombre:String",  
    "username": "username:String",  
    "phone": "telefono:String"  
  },  
  "code": "SUCCESS (0) || ERROR (1)",  
  "message": "STRING VACÍO || MENSAJE DE ERROR"  
}
```


Consultar contactos

URL: `http://[BASE_URL]/services/mybets_consult_contacts`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}
```

Respuesta:

```
{
  "data": [{
    "username": "username:String",
    "id": "id:Integer",
    "email": "email:String",
    "phone": "telefono:String",
    "name": "nombre:String",
    "imagen": "URLimagen:String"
  },
  {
    "username": "username:String",
    "id": "id:Integer",
    "email": "email:String",
    "phone": "telefono:String",
    "name": "nombre:String",
    "imagen": "URLimagen:String"
  }
],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Consultar últimas partidas

URL: `http://[BASE_URL]/services/mybets_ultimas_partidas`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}
```

Respuesta:

```
{
  "data": [{
    "idpartida": "id:Integer",
    "nombrePartida": "nombre:String",
    "fecha": "fecha:String(dd/MM/yyyy HH:mm)",
    "bote": "bote:Integer",
    "numpersonas": "numpersonas:Integer",
    "urlicono": "urlicono:String",
    "coloricono": "colorrgb:String(#000000)",
    "tipopartida": "1:Integer ResultadoExacto || 0:Integer 1x2",
    "participa": "0:Integer NO || 1:Integer SI"
  }, {
    "idpartida": "id:Integer",
    "nombrepartida": "nombre:String",
    "fecha": "fecha:String(dd/MM/yyyy HH:mm)",
    "bote": "bote:Integer",
    "numpersonas": "numpersonas:Integer",
    "urldelicono": "urldelicono:String",
    "coloricono": "color:String(#000000)",
  }
]
```

```

    "tipopartida": "0:INT 1x2 || 1:INT result ex",
    "participa": "0:INT NO || 1:INT SI"
  }],
  "idfutbol": "id:String",
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```

```

tipopartida:Integer:
0 = Partida tipo 1x2
1 = Partida tipo Resultado Exacto

```

Consultar categorias

URL: http://[BASE_URL]/services/mybets_consult_categories

Método: POST

Parámetros:

```

{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}

```

Respuesta:

```

{
  "data": [{
    "icon": "URLImagen:String",
    "id": "id:Integer",
    "name": "nombre:String"
  }],
  {
    "icon": "URLImagen:String",

```

```
    "id": "id:Integer",
    "name": "nombre:String"
  },...],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Invitar a un amigo a la aplicación

URL: http://[BASE_URL]/services/mybets_peticion_amistad

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idamigo": "idamigo:Integer"
  }
}
```

Respuesta:

```
{
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Crear una partida

URL: `http://[BASE_URL]/services/mybets_crear_partida`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "idpartida:Integer",
    "idamigos": [idamigo:Integer, idamigo:Integer, ...]
  }
}
```

Respuesta:

```
{
  "idjornada": "ID:Integer",
  "idpartida": "ID:Integer",
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Aceptar partida

URL: `http://[BASE_URL]/services/mybets_aceptar_partida`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "idpartida:Integer"
  }
}
```

Respuesta:

```
{
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Realizar apuesta (1x2, Exacta,...)

URL: http://[BASE_URL]/services/mybets_realizar_apuesta

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "idpartida:Integer",
    "idjornada": "idjornada:Integer",
    "apuestas": [apuesta1:Integer, apuesta2:Integer,...]
  }
}
```

Respuesta:

```
{
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Consultar Perfil

URL: `http://[BASE_URL]/services/mybets_perfil`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}
```

Respuesta:

```
{
  "user": {
    "nombre" :
      "nombre:String",
      "id": "id:Integer",
      "puntosTotales": "puntos:Integer",
      "imagen" : "URLImagen:String"
    },
  "data": [
    {
      "nombre": "nombrepartida:String",
      "puntos": "puntosganados:Integer",
      "fecha": "fecha:String"
    },
    {
      "nombre": "nombrepartida:String",
      "puntos": "puntosganados:Integer",
      "fecha": "fecha:String"
    }
  ]
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Consultar Partidas de una categoría

URL: [http://\[BASE_URL\]/services/mybets_partidas_categoria](http://[BASE_URL]/services/mybets_partidas_categoria)

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idcategoria": "idcategoria:Integer"
  }
}
```

Respuesta:

```
{
  "subcategorias": [{
    "idsubcategoria": "id:Integer",
    "nombresubcategoria": "nombre:String",
    {...}],
  "data": [
    {
      "idpartida": "id:Integer",
      "nombrePartida": "nombre:String",
      "fecha": "fecha:String",
      "bote": "bote:Integer",
      "numpersonas": "numpersonas:Integer",
      "urlicono": "URLIcono:String",
      "tipopartida": "0:Integer 1x2 | 1:Integer result ex",
      "participa": "0:Integer NO | 1:Integer SI"
    },
    {
      "idpartida": "id:Integer",
      "nombrePartida": "nombre:String",
      "fecha": "fecha:String",
      "bote": "bote:Int",
      "numpersonas": "numpersonas:Integer",

```



```

    "urlicono": "URLIcono:String",
    "tipopartida": "0:Integer 1x2 | 1:Integer result ex",
    "participa": "0:Integer NO | 1:Integer SI"
  }...],
  "color": "color:String(#000000)",
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```

Consultar Partidas Pendientes

URL: [http://\[BASE_URL\]/services/mybets_partidas_pendientes](http://[BASE_URL]/services/mybets_partidas_pendientes)

Método: POST

Parámetros:

```

{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}

```

Respuesta:

```

{
  "data": [
    {
      "idpartida": "id:Integer",
      "nombrepartida": "nombre:String",
      "fecha": "fecha:String",
      "bote": "bote:Int",
      "numpersonas": "numpersonas:Integer",
      "urlicono": "URLIcono:String",
      "coloricono": "color:String(#454545)",
      "tipopartida": "0:Integer 1x2 | 1:Integer result ex",
      "estadoapuesta": "0:Integer = rechazada, 1:Integer = esperando"
    }
  ]
}

```

```

    resultado, 2:Integer = jugar ya",
    "participa": "0:Integer NO | 1:Integer SI"
  },
  {
    "idpartida": "id:Integer",
    "nombrepartida": "nombre:String",
    "fecha": "fecha:String",
    "bote": "bote:Int",
    "numpersonas": "numpersonas:Integer",
    "urlicono": "URLIcono:String",
    "coloricono": "color:String(#454545)",
    "tipopartida": "0:Integer 1x2 | 1:Integer result ex",
    "estadoapuesta": "0:Integer = rechazada, 1:Integer = esperando
    resultado, 2:Integer = jugar ya",
    "participa": "0:Integer NO | 1:Integer SI"
  }...]
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```

Consultar Partidas Pasadas

URL: http://[BASE_URL]/services/mybets_partidas_pasadas

Método: POST

Parámetros:

```

{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}

```

Respuesta:

```
{
  "data": [
    {
      "idpartida": "id:Integer",
      "nombrepartida": "nombre:String",
      "puntosganados": "puntos:Integer",
      "bote": "bote:Integer",
      "numpersonas": "numpersonas:Integer",
      "urlicono": "URLIcono:String",
      "coloricono": "color:String(#454545)",
      "tipopartida": "0:Integer 1x2 | 1:Integer result ex"
    },
    {
      "idpartida": "id:Integer",
      "nombrepartida": "nombre:String",
      "puntosganados": "puntos:Integer",
      "bote": "bote:Integer",
      "numpersonas": "numpersonas:Integer",
      "urlicono": "URLIcono:String",
      "coloricono": "color:String(#454545)",
      "tipopartida": "0:Integer 1x2 | 1:Integer result ex"
    },... ]
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Consultar Muro

URL: `http://[BASE_URL]/services/mybets_consultar_muro`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}
```

Respuesta:

```
{
  "data": [
    {
      "nombreamigo": "nombre:String",
      "puntosganados": "puntos:Integer",
      "nombrejuego": "nombrejuego:String",
      "fecha": "fecha:String",
      "urlfoto": "URLFoto:String"
    },
    {
      "nombreamigo": "nombre:String",
      "puntosganados": "puntos:Integer",
      "nombrejuego": "nombrejuego:String",
      "fecha": "fecha:String",
      "urlfoto": "URLFoto:String"
    },
    ...
  ],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Cambiar Nombre Partida

URL: [http://\[BASE_URL\]/services/mybets_cambiar_nombre_partida](http://[BASE_URL]/services/mybets_cambiar_nombre_partida)

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "id:Integer",
    "nuevonombre": "nombrenuevo:String"
  }
}
```

Respuesta:

```
{
  "data": [
    {
      "idpartida": "id:Integer",
      "nombrepartida": "nombre:String",
      "fecha": "fecha:String",
      "bote": "bote:Integer",
      "numpersonas": "numpersonas:Int",
      "urlicono": "URLIcono:String",
      "coloricono": "color:String(#454545)",
      "tipopartida": "0:Integer 1x2 | 1:Integer result ex",
      "estadoapuesta": "0 = rechazada, 1 = esperando resultado, 2 = jugar ya",
      "participa": "0:INTEGER NO | 1:INTEGER SI"
    },
    {
      "idpartida": "id:Integer",
      "nombrepartida": "nombre:String",
      "fecha": "fecha:String",
      "bote": "bote:Integer",
    }
  ]
}
```

```

    "numpersonas": "numpersonas:Integer",
    "urlicono": "URLIcono:String",
    "coloricono": "color:String(#454545)",
    "tipopartida": "0:INTEGER 1x2 | 1:INTEGER result ex",
    "estadoapuesta": "0 = rechazada, 1 = esperando resultado, 2 =
jugar ya",
    "participa": "0:INTEGER NO | 1:INTEGER SI"
  },...],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```

Rechazar Partida

URL: http://[BASE_URL]/services/mybets_rechazar_partida

Método: POST

Parámetros:

```

{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "id:Integer"
  }
}

```

Respuesta:

```

{
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```

Consultar Partidas de una subcategoría

URL: [http://\[BASE_URL\]/services/mybets_partidas_subcategoria](http://[BASE_URL]/services/mybets_partidas_subcategoria)

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idsubcategoria": "id:Integer"
  }
}
```

Respuesta:

```
{
  "data": [
    {
      "idpartida": "id:Integer",
      "nombrepartida": "nombre:String",
      "fecha": "fecha:String",
      "bote": "bote:Integer",
      "numpersonas": "numpersonas:Int",
      "urlicono": "URLIcono:String",
      "coloricono": "color:String(#454545)",
      "tipopartida": "0:Integer 1x2 | 1:Integer result ex",
      "participa": "0:Integer NO | 1: Integer SI"
    },
    {
      "idpartida": "id:Integer",
      "nombrepartida": "nombre:String",
      "fecha": "fecha:String",
      "bote": "bote:Int",
      "numpersonas": "numpersonas:Integer",
      "urlicono": "URLIcono:String",
      "coloricono": "color:String(#454545)",
      "tipopartida": "0:Integer 1x2 | 1:Integer result ex",
      "participa": "0:Integer NO | 1:Integer SI"
    }
  ]
}
```

```
},...],  
"code": "SUCCESS (0) || ERROR (1)",  
"message": "STRING VACÍO || MENSAJE DE ERROR"  
}
```

Consultar Datos Usuario

URL: http://[BASE_URL]/services/mybets_consultar_datos_usuario

Método: POST

Parámetros:

```
{  
  "request": {  
    "email": "email:String",  
    "password": "password:String(md5)"  
  }  
}
```

Respuesta:

```
{  
  "data": {  
    "username": "username:String",  
    "name": "nombre:String",  
    "email": "email:String",  
    "imagen": "URLImagen:String"  
  },  
  "code": "SUCCESS (0) || ERROR (1)",  
  "message": "STRING VACÍO || MENSAJE DE ERROR"  
}
```


Modificar Datos Usuario

URL: http://[BASE_URL]/services/mybets_modificar_datos_usuario

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "username": "username:String",
    "name": "nombre:String",
    "image" : "imagen:Base64String"
  }
}
```

Respuesta:

```
{
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Pedir Datos Jornada

URL: http://[BASE_URL]/services/mybets_pedir_datos_jornada

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "ID:Integer"
  }
}
```

Respuesta:

```
{
  "data" : {
    "idjornada": "ID:INTEGER",
    "listapartidos": [
      { "idpartido" : "ID:INTEGER",
        "listaequipos" : [{ "id" : "ID:INTEGER",
                           "nombre" : "NOMBRE:STRING",
                           "iconurl" : "URL:STRING" }, ... ]
      }, ... ]
    },
    "code": "SUCCESS (0) || ERROR (1)",
    "message": "STRING VACÍO || MENSAJE DE ERROR"
  }
}
```

Pedir Clasificación de una partida

URL: [http://\[BASE_URL\]/services/mybets_clasificacion](http://[BASE_URL]/services/mybets_clasificacion)

Método: POST

Parámetros:

```
{
  "request": {
    "email": "EMAIL:STRING",
    "password": "ENCRYPTED_PASSWORD:STRING",
    "idpartida": "ID:INTEGER"
  }
}
```

Respuesta:

```
{
  "partida" : { "nombre": "NOMBRE PARTIDA",
                "fecha": "FECHA:STRING",
                "bote": "BOTE:INTEGER",
```

```

        "urlicono": "URLICONO",
        "colorfondo": "COLOR=#454545"},
    "data" : [
    {
        "idusuario": "ID",
        "nombre": "NOMBRE:STRING",
        "puntos": "PUNTOS:INTEGER",
        "urlicono": "URLICONO:STRING",
        "posicion": "NUMERO:INTEGER"
    },
    {
        "idusuario": "ID",
        "nombre": "NOMBRE",
        "puntos": "PUNTOS:INTEGER",
        "urlicono": "URLICONO:STRING",
        "posicion": "NUMERO:INTEGER"
    }...],
    "code": "SUCCESS (0) || ERROR (1)",
    "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```

Consultar Partidas Populares

URL: http://[BASE_URL]/services/mybets_partidas_populares

Método: POST

Parámetros:

```

{
    "request": {
        "email": "email:String",
        "password": "password:String(md5)"
    }
}

```

Respuesta:

```
{
  "data": [
    {
      "idpartida": "ID",
      "nombrepartida": "NOMBRE",
      "fecha": "FECHA",
      "bote": "BOTE",
      "numpersonas": "NUMPERS",
      "urlicono": "URLICONO",
      "coloricono": "COLORRGB:#454545",
      "tipopartida": "0:INTEGER 1x2 | 1:INTEGER result ex",
      "participa": "0:INTEGER NO | 1:INTEGER SI"
    },
    {
      "idpartida": "ID",
      "nombrepartida": "NOMBRE",
      "fecha": "FECHA",
      "bote": "BOTE",
      "numpersonas": "NUMPERS",
      "urlicono": "URLICONO",
      "coloricono": "COLORRGB:#454545",
      "tipopartida": "0:INTEGER 1x2 | 1:INTEGER result ex",
      "participa": "0:INTEGER NO | 1:INTEGER SI"
    },
    ...
  ],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Obtener Puntuación Usuario

URL: [http://\[BASE_URL\]/services/mybets_puntuacion](http://[BASE_URL]/services/mybets_puntuacion)

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}
```

Respuesta:

```
{
  "data": {
    "puntuacion": "PUNTUACION:INTEGER"
  },
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Obtener subcategoría Más Usada

URL: [http://\[BASE_URL\]/services/mybets_subcategoria_mas_usada](http://[BASE_URL]/services/mybets_subcategoria_mas_usada)

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)"
  }
}
```

Respuesta:

```
{
  "data": {
    "nombresubcategoria": "nombre:String",
    "idsubcategoria": "id:Integer"
  },
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Obtener Apuestas Realizadas (1x2)

URL: [http://\[BASE_URL\]/services/mybets_apuestas_realizadas](http://[BASE_URL]/services/mybets_apuestas_realizadas)

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "ID:Integer"
  }
}
```

Respuesta:

```
{
  "partida" : { "fecha": "FECHA:STRING",
    "bote": "BOTE:INTEGER"
    "urlicono": "URLICONO",
    "colorfondo": "COLOR=#454545"} ,
  "data": [
    {
      "urlperfil": "URLImagenPerfil:String",
      "idusuario": "idusuario:Integer",
      "apuestas": [{
        "nombrelocal": "nombre:String",
        "urllocal": "URLImagen:String",
        "nombrevisitante": "nombre:String",
        "urlvisitante": "URLImagen:String",
        "apuesta": "apuesta:Integer"} ,
        {"nombrelocal": "nombre:String",
        "urllocal": "URLImagen:String",
        "nombrevisitante": "nombre:String",
        "urlvisitante": "URLImagen:String",
        "apuesta": "apuesta:Integer"} , ...]
    },
    {
      "urlperfil": "URLImagenPerfil:String",
      "idusuario": "idusuario:Integer",
```

```

    "apuestas": [{
      "nombrelocal": "nombre:String",
      "urllocal": "URLImagen:String",
      "nombrevisitante": "nombre:String",
      "urlvisitante": "URLImagen:String",
      "apuesta": "apuesta:Integer"},
      {"nombrelocal": "nombre:String",
      "urllocal": "URLImagen:String",
      "nombrevisitante": "nombre:String",
      "urlvisitante": "URLImagen:String",
      "apuesta": "apuesta:Integer" },...]
  },...],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```

Obtener Apuestas Realizadas (Exacto)

URL: [http://\[BASE_URL\]/services/mybets_apuestas_realizadas](http://[BASE_URL]/services/mybets_apuestas_realizadas)

Parámetros:

```

{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idpartida": "ID:Integer"
  }
}

```

Respuesta:

```

{
  "partida" : { "fecha": "FECHA:STRING",
    "bote": "BOTE:INTEGER",
    "urlicono": "URLICONO",
    "colorfondo": "COLOR=#454545" },

```

```

"data": [
{
    "urlperfil": "URLImagenPerfil:String",
    "idusuario": "id:Integer",
    "apuestas": [{
        "nombrelocal": "nombre:String",
        "urllocal": "URLImagen:String",
        "nombrevisitante": "nombre:String",
        "urlvisitante": "URLImagen:String",
        "apuesta": [ScoreEquipo1:Integer, ScoreEquipo2:Integer]},
        {"nombrelocal": "nombre:String",
        "urllocal": "URLImagen:String",
        "nombrevisitante": "nombre:String",
        "urlvisitante": "URLImagen:String",
        "apuesta": [ScoreEquipo1:Integer, ScoreEquipo2:Integer] },...]
},
{
    "urlperfil": "URLImagenPerfil:String",
    "idusuario": "id:Integer",
    "apuestas": [{
        "nombrelocal": "nombre:String",
        "urllocal": "URLImagen:String",
        "nombrevisitante": "nombre:String",
        "urlvisitante": "URLImagen:String",
        "apuesta": [ScoreEquipo1:Integer, ScoreEquipo2:Integer]},
        {"nombrelocal": "nombre:String",
        "urllocal": "URLImagen:String",
        "nombrevisitante": "nombre:String",
        "urlvisitante": "URLImagen:String",
        "apuesta": [ScoreEquipo1:Integer, ScoreEquipo2:Integer] },...]
},...],
    "code": "SUCCESS (0) || ERROR (1)",
    "message": "STRING VACÍO || MENSAJE DE ERROR"
}

```


Consultar Perfil amigo

URL: `http://[BASE_URL]/services/mybets_perfil_amigo`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "idamigo": "id:Integer"
  }
}
```

Respuesta:

```
{
  "user": {
    "nombre": "nombre:String",
    "id": "id:Int",
    "puntosTotales": "puntos:Int",
    "imagen": "URLImagen:String"
  },
  "data": [
    {
      "nombre": "nombre:String",
      "puntos": "puntos:Int",
      "fecha": "fecha:String"
    },
    {
      "nombre": "nombre:String",
      "puntos": "puntos:Int",
      "fecha": "fecha:String"
    }
  ],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Comprobar lista amigos

URL: `http://[BASE_URL]/services/mybets_lista_amigos`

Método: POST

Parámetros:

```
{
  "request": {
    "email": "email:String",
    "password": "password:String(md5)",
    "emails": [email:String, ... ],
    "telefonos": [telefono:String, ...]
  }
}
```

Respuesta:

```
{
  "data": [
    {
      "nombre": "nombre:String",
      "urlicono": "URLIcono:String",
      "idusuario": "id:Integer",
      "telefono": "telefono:String",
      "email": "email:String"
    },
    {
      "nombre": "nombre:String",
      "urlicono": "URLIcono:String",
      "idusuario": "id:Integer",
      "telefono": "telefono:String",
      "email": "email:String"
    },
    ... ],
  "code": "SUCCESS (0) || ERROR (1)",
  "message": "STRING VACÍO || MENSAJE DE ERROR"
}
```

Anexo B

Glosario

Bundle: conjunto estructurado de archivos que se encuentran en un directorio e implementan una característica.

CSS: conjunto de reglas que definen el formato de presentación de los elementos de una página HTML.

JSON: formato de texto ligero orientado al intercambio de datos.

Puntos de historia: medida de esfuerzo asociada a los puntos de historia.

REST: tipo de arquitectura de desarrollo web que permite crear servicios y aplicaciones que pueden ser utilizados por cualquier dispositivo o cliente que entienda HTTP.

Website: se refiere al sitio web, colección de páginas web relacionadas típicamente identificadas bajo un dominio común y publicadas al menos en un servidor web.